WHAT IS CLAIMED IS:

1.An apparatus for generating a finite state transducer for use in incremental parsing, comprising:

a recursive transition network creating device that creates a recursive transition network, the recursive transition network being a set of networks, each network representing a set of grammar rules based on a context-free grammar by states and arcs connecting the states, each arc having an input label and an output label, each network having a recursive structure where each transition labeled with a non-terminal symbol included in each of the networks is defined by another network;

an arc replacement device that replaces an arc having an input label representing a start symbol included in the finite state transducer in an initial state by a network corresponding to the input label of the arc in the recursive transition network and further recursively repeats an arc replacement operation for replacing each arc, which is newly created from a replaced network, by another network in the recursive transition network; and

a priority calculating device that calculates a derivation probability to derive a node of a parse tree corresponding to each of arcs whose input labels are non-terminal symbols in the finite state transducer based on statistical information regarding frequency of applying grammar rules and determines an arc replacement priority in terms of an obtained derivation probability;

wherein the arc replacement device continues applying the arc replacement operation to each arc included in the finite state transducer in descending order of the arc replacement priority until the finite state transducer reaches a predetermined size.

2.The apparatus according to claim 1, further comprising an arc eliminating device that, after the application of the arc replacement operation by the arc replacement device terminates, eliminates arcs whose input labels are non-terminal symbols and further performs the arc replacement operation.

3.The apparatus according to claim 1, wherein the derivation probability for a certain node represents a probability that grammar rules are applied in order to each node on a path from a root node to the certain node in the parse tree.

4.The apparatus according to claim 3, wherein derivation probability P $(Xr_{M(1M)})$ for node $Xr_{M(1M)}$ is determined as follows:

$$P\left(X_{r_{M(l_M)}}\right) = \prod_{i=1}^{M} \hat{P}\left(r_i \mid r_{i-N+1(l_{i-N+1})}, \ldots, r_{i-1(l_{i-1})}\right)$$

wherein $r_i$ represents a grammar rule, $r_i(l_i)$ represents that grammar rule $r_i$ is applied and grammar rule $r_i$+1 to be applied next is applied to a node generated by the $(l_i)$-th element of the right side of $r_i$, and N is a predetermined positive integer.

5. A computer-readable recording medium storing a program for generating a finite state transducer for use in incremental parsing, the program comprising:

a recursive transition network creating routine that creates a recursive transition network, the recursive transition network being a set of networks, each network representing a set of grammar rules based on a context-free grammar by states and arcs connecting the states, each arc having an input label and an output label, each network having a recursive structure where each transition labeled with a non-terminal symbol included in each of the networks is defined by another network;

an arc replacement routine that replaces an arc having an input label representing a start symbol included in the finite state transducer in an initial state by a network corresponding to the input label of the arc in the recursive transition network and further recursively repeats an arc replacement operation for replacing each arc, which is newly created from a replaced network, by another network in the recursive transition network; and

a priority calculating routine that calculates a derivation probability to derive a node of a parse tree corresponding to each of arcs whose input labels are non-terminal symbols in the finite state transducer based on statistical information regarding frequency of applying grammar rules and determines an arc replacement priority in terms of an obtained derivation probability;

wherein the arc replacement routine continues applying the arc replacement operation to each arc included in the finite state transducer in descending order of the arc replacement priority until the finite state transducer reaches a predetermined size.

6. The computer-readable recording medium according to claim 5, the program further comprising an arc eliminating routine that, after the application of the arc replacement operation by the arc replacement routine terminates, eliminates arcs whose input labels are non-terminal symbols and further performs the arc replacement operation.

7. The computer-readable recording medium according to claim 5, wherein, in the program, the derivation probability for a certain node represents a probability that grammar rules are applied in order to each node on a path from a root node to the certain node in the parse tree.

8. The computer-readable recording medium according to claim 7, wherein derivation probability $P\ (Xr_{M(1M)})$ for node $Xr_{M(1M)}$ is determined as follows:

$$P(X_{r_{M(l_M)}}) = \prod_{i=1}^{M} \hat{P}(r_i \mid r_{i-N+1(l_{i-N+1})}, \ldots, r_{i-1(l_{i-1})})$$

wherein $r_i$ represents a grammar rule, $r_i(l_i)$ represents that grammar rule $r_i$ is applied and grammar rule $r_i$+1 to be applied next is applied to a node generated by the $(l_i)$-th element of the right side of $r_i$, and N is a predetermined positive integer.

9. A method for generating a finite state transducer for use in incremental parsing comprising the steps of:

creating a recursive transition network, the recursive transition network being a set of networks, each network representing a set of grammar rules based on a context-free grammar by states and arcs connecting the states, each arc having an input label and an output label, each network having a recursive structure where each transition labeled with a non-terminal symbol included in each of the networks is defined by another network;

replacing an arc having an input label representing a start symbol included in the finite state transducer in an initial state by a network corresponding to the input label of the arc in the recursive transition network and further recursively repeating an arc replacement operation for replacing each arc, which is newly created from a replaced network, by another network in the recursive transition network; and

calculating a derivation probability to derive a node of a parse tree corresponding to each of arcs whose input labels are non-terminal symbols in the finite state transducer based on statistical information regarding frequency of applying grammar rules and determines an arc replacement priority in terms of an obtained derivation probability;

wherein, in the step of replacing an arc, continuing applying the arc replacement operation to each arc included in the finite state transducer in descending order of the arc replacement priority until the finite state transducer reaches a predetermined size.

10. The method according to claim 9, further comprising the step of eliminating arcs whose input labels are non-terminal symbols and further performs the arc replacement operation, after the application of the arc replacement operation by the arc replacement device terminates.

11. The method according to claim 9, wherein the derivation

probability for a certain node represents a probability that grammar rules are applied in order to each node on a path from a root node to the certain node in the parse tree.

12. The method according to claim 11, wherein derivation probability $P\ (Xr_{M(1M)})$ for node $Xr_{M(1M)}$ is determined as follows:

$$P(X_{r_{M(l_M)}}) = \prod_{i=1}^{M} \hat{P}(r_i \mid r_{i-N+1(l_{i-N+1})}, \ldots, r_{i-1(l_{i-1})})$$

wherein $r_i$ represents a grammar rule, $r_i(l_i)$ represents that grammar rule $r_i$ is applied and grammar rule $r_i+1$ to be applied next is applied to a node generated by the $(l_i)$-th element of the right side of $r_i$, and N is a predetermined positive integer.

13. An apparatus for incremental parsing, comprising:

a finite state transducer generated by the method according to claim 7, the finite state transducer outputting one or more pieces of a parse tree as a result of a state transition when each word is inputted thereto; and

a connecting device that sequentially connects each piece of the parse tree outputted by the finite state transducer.